

JANEALYSSON DOS SANTOS DE ARAUJO

## MÉTRICAS PARA DEFINIR O TAMANHO DE UM SISTEMA

Artigo apresentado como requisito parcial à obtenção do grau de especialista em Desenvolvimento de Softwares em Mercados Internacionais.

Orientador: prof. Dr. Celso Yoshikazu Ishida

CURITIBA

2012

## RESUMO

Métricas de software são padrões quantitativos de medidas que contribuem para a estimativa de prazo e custos mais precisos de projetos ou produtos de softwares. São utilizadas como apoio nas boas práticas do gerenciamento de projetos facilitando o processo na tomada de decisões e a subsequente obtenção nas medidas de produtividade e qualidade. A Análise de Ponto de Função (APF) e Pontos por Caso de Uso (PCU) são duas métricas muito utilizadas na estimativa do tamanho de sistemas mas cada uma possui características específicas que as tornam distintas. A APF pode ser estimada com diversos documentos criados na fase de análise de requisitos além da própria aplicação e o método para quantificação do sistema é mais trivial do que o método para quantificação do PCU que necessita dos Casos de Uso atualizados. Este artigo apresenta um comparativo entre duas métricas para quantificar o tamanho de um sistema no início de um projeto auxiliando os desenvolvedores e gerentes de projetos na extração de requisitos e estimativa de esforço.

**Palavras-Chaves:** Métricas, tamanho, software, APF, PCU.

## ABSTRACT

Software metrics are standard quantitative measures that contribute to the estimated costs and more accurate term projects or software products. Metrics are used to support the practice of project management by facilitating the decision making process and the subsequent obtaining the measures of productivity and quality. The Function Point Analysis (FPA) and by Use Case Points (PCU) are two widely used metrics in estimating the size of systems but each has specific characteristics that make them distinct. The APF can be estimated with several documents created during requirements analysis beyond the application itself and the method for quantification of the system is more than the trivial method to quantify the PCU it needs from Use Cases updated. This paper presents a comparison between two metrics to quantify the size of a system at the beginning of a project helping developers and project managers in extracting requirements and effort estimation.

**Keywords:** Metrics, size, software, FPA, UCP.

## INTRODUÇÃO

A medição de software tem-se convertido em uma parte essencial da Engenharia de Software. Desenvolvedores de sistemas medem determinadas características do software para saber se os requisitos são consistentes e completos, se os projetos estão corretamente documentados ou se o código está pronto para ser testado. Gerentes de projeto medem atributos de processos e produtos para decidirem quando o software poderá ser entregue e se o projeto será executado dentro do prazo. (FENTON, 1996).

Métricas de software, portanto, são padrões quantitativos de medidas de vários aspectos de um projeto ou produto de software, e constituem-se em uma poderosa ferramenta gerencial, contribuindo para a elaboração de estimativas de prazo e custo mais precisas e para o estabelecimento de metas plausíveis, facilitando assim o processo de tomada de decisões e a posterior obtenção de medidas de produtividade e qualidade. Em geral, a aplicação de métricas é muito importante porque favorece três atividades básicas: (BERTOLAMI, 2003)

1. entendimento do que está ocorrendo durante o processo de desenvolvimento ou manutenção, o que permite estabelecer parâmetros de controle para futuros comportamentos;

2. controle do andamento de projetos com base em metas pré-estabelecidas ou dados históricos, permitindo a rápida atuação no sentido de corrigir eventuais desvios;

3. melhoria dos processos e produtos como, por exemplo, o aumento da quantidade ou tipo de revisões de projeto baseada nas medidas de qualidade da especificação.

Podemos dividir as métricas de software em medidas diretas relacionadas ao custo, esforço, quantidade de linhas de código e quantidade de erro, e indiretas relacionadas a funcionalidade, qualidade, complexidade, eficiência e confiabilidade.

Existem dificuldades sobre o que deve ser medido e como devem ser interpretados os resultados que são utilizados para analisar a qualidade, produtividade do processo de desenvolvimento e manutenção, qualificação do desempenho técnico da perspectiva do desenvolvedor, desempenho do produto da

perspectiva do usuário independente das tecnologias utilizadas e comparação da produtividade entre tecnologias diferentes.

As estimativas que não possuem um padrão de medição impactam em atividades não realizadas, deficiência funcional do produto, custos elevados e atrasos na entrega.

Diversas métricas para medir o tamanho de um sistema estão disponíveis, mas todas possuem limitações e requisitos que podem dificultar a sua utilização. Este trabalho tem por objetivo realizar uma análise das características entre os métodos de APF e PCU permitindo avaliar o grau aplicabilidade e contribuição para estimativa de esforço.

## Métricas de Software

Durante os 10 primeiros anos ou mais da indústria de software, iniciados por volta de 1947 a 1957, a maioria das aplicações eram muito pequenas. A grande maioria era menor que 1.000 linhas de código fonte. Todos foram escritos em linguagens de baixo nível e alguns foram corrigidos em linguagem de máquina, o que é ainda mais difícil de trabalhar. (JONES, 2008).

A primeira tentativa de medir a produtividade e a qualidade usando a métrica de "linhas de código" foi na época de 1950 sendo bastante eficaz. A codificação levou cerca de 50 por cento do esforço para construir uma aplicação, a depuração e testes levaram cerca de 40 por cento e o restante levou apenas cerca de 10 por cento. (JONES, 2008).

Antigamente a produtividade era baseada em linhas de código e a contagem de bugs baseada em erros ou defeitos por 1.000 linhas de código ( KLOC ) sendo o padrão de métricas e bastante eficaz. Isto ocorria porque os erros eram mais comuns e problemáticos nos primeiros dias de codificação do software. (JONES, 2008).

Entre 1957 e 1967, a situação começou a mudar drasticamente. As linguagens de baixo nível começaram a ser substituídas por linguagens mais poderosas, como COBOL, FORTRAN, APL e o tamanho das aplicações cresceram de 1.000 linhas de código para 100 mil linhas de código. (JONES, 2008).

Estas mudanças nas linguagens de programação e tamanhos das aplicações começaram a causar problemas para a métrica de linhas de código. Em 1967, a codificação em si estava começando a cair abaixo de 30 por cento do esforço de desenvolvimento, enquanto a elaboração de requisitos, especificações, planos e documentação começaram a se aproximar de 40 por cento. Testar e depurar levou cerca de 30 por cento. Deste modo, alguns aplicativos foram escritos em duas ou mais linguagens de programação diferentes. (JONES, 2008).

Em meados da década de 1970 foram notados problemas mais sérios com a métrica LOC. É razoável supor que as linguagens de programação de alto nível melhoraram a produtividade do desenvolvimento e da qualidade. (JONES, 2008).

Suponha que você tenha duas aplicações idênticas, sendo uma codificada em Assembly e outra em COBOL. A versão em Assembly possui 10.000 linhas de

código, mas a versão em COBOL possui apenas 3.000 linhas de código. Quando você mede as taxas de codificação, ambas as línguas foram codificadas a uma taxa de 1.000 linhas de código por mês, mas uma vez que a versão em COBOL era apenas um terço do tamanho da versão em Assembly, os ganhos de produtividade não podem ser vistos por meio de métricas LOC. (JONES, 2008).

Presumimos que as especificações e documentos para ambas as versões levaram 5 meses de esforço. Assim, a versão em Assembly necessitou de 15 meses de esforço enquanto a versão em COBOL precisaria de apenas 8 meses de esforço. Se medir todo o projeto com métricas LOC, a versão em Assembly tem uma taxa de produtividade de 666 linhas de código por mês enquanto a versão em COBOL tem uma produtividade de 375 linhas de código por mês. Obviamente, os benefícios econômicos de linguagens de programação de alto nível desaparecem quando a métrica LOC era utilizada. (JONES, 2008).

Estes problemas incentivaram a pesquisa e desenvolvimento de novas métricas que mediriam os sistemas independente do volume de código, produtividade e qualidade sem distorção.

## **Análise de Pontos de Função**

A Análise de Pontos de Função (APF) é uma medida de dimensionamento de significância de negócio claro. Tornou-se pública por Allan Albrecht da IBM em 1979, a técnica APF quantifica as funções contidas no software em termos que sejam significativos para os usuários do software. A medida se relaciona diretamente com os requisitos de negócio que o software se destina a abordar. Pode ser facilmente aplicada através de uma ampla variedade de ambientes de desenvolvimento e ao longo da vida de um projeto de desenvolvimento. Medidas de outras empresas, tais como a produtividade do processo de desenvolvimento e os custos por unidade de suporte de software, também pode ser medida em pontos prontamente derivados. A função é derivada de uma série de etapas. Usando um conjunto padronizado de critérios básicos, cada uma das funções do negócio é um índice numérico de acordo com o seu tipo e complexidade. Estes índices são somados para obter uma primeira medida de tamanho o qual é então normalizado pela incorporação de um número de fatores relacionados com o software como um

todo. O resultado final é um número único, denominado o índice de Ponto de Função que mede o tamanho e complexidade do produto de software. (IFPUG, 2012).

A APF Métricas (2012) afirma que o Brasil, atualmente, é o país com o maior número de certificados do mundo, totalizando aproximadamente 200 certificados, do total de 600 no mundo. As principais empresas que utilizam APF em seus contratos são BANDES, Caixa Econômica Federal, Serpro, PM-SP, Dataprev, Correios, Bradesco, Petrobrás, Embratel, Prodeb, EDS, Politec, MSAINFOR, DBA, CPMBRAXIS, ABN/REAL, Stefanini, TAM, Oi, Scopus, GPTI, BRQ, DTS Consulting, Infoserver, 7COMM, Prime Informática, Sistran, Teradata, Procwork, TIVIT, Getronic, HST e T-SYSTEM.

O Manual de Práticas de Contagem de Pontos de Função contém todas as definições e regras necessárias para o uso da APF, disponibiliza exemplos mais comuns e aborda de maneira geral o uso da técnica para que o processo seja aplicado em uma grande variedade de situações (VAZQUEZ, 2012).

Segundo Vazquez (2012) ao utilizar a APF é possível medir a funcionalidade que o usuário solicita e recebe e medir o desenvolvimento e a manutenção de software, independentemente da tecnologia utilizada para implementação.

A desvantagem de utilizar APF é de não medir diretamente o esforço, a produtividade ou o custo. É exclusivamente uma medida de tamanho funcional do software. Este tamanho, junto com outras variáveis, é que pode ser usado para derivar produtividade, esforço e custo do projeto de software.

A Figura 1 representa todos os componentes utilizados durante o processo da contagem de pontos de função que estão distribuídos em Funções de Dados e Funções de Transação, respectivamente, funções que representam grupos lógicos e funções que representam os tipos de processos.

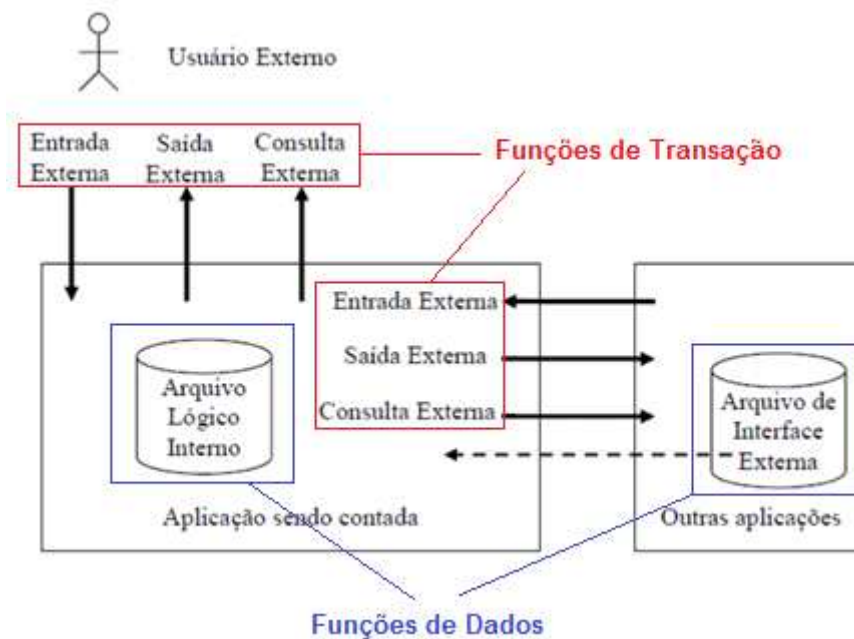


Figura 1 - Componentes da Contagem ( Adaptado de IFPUG, 2010 )

As Funções de Dados distribuem-se em Arquivo Lógico Interno (ALI) e Arquivo Lógico Externo (ALE), respectivamente, um grupo de dados logicamente relacionados, reconhecido pelo usuário, mantido por meio de um processo elementar da aplicação que está sendo contada e um grupo de dados, logicamente relacionados, reconhecido pelo usuário, mantido por meio de um processo elementar de uma outra aplicação e referenciado pela aplicação que está sendo contada. O ALE é obrigatoriamente um ALI de outra aplicação.

As Funções de Transação constituem em Entrada Externa (EE), Consulta Externa (CE) e Saída Externa (SE), respectivamente, um processo elementar que processa dados ou informação de controle que entram pela fronteira da aplicação com o objetivo principal de manter um ou mais ALI ou alterar o comportamento do sistema, um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação com o objetivo principal de apresentar informação para o usuário através da recuperação de dados ou informação de controle de ALI ou ALE e um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação com o objetivo principal de apresentar informação para um usuário ou outra aplicação através de um processamento lógico adicional à recuperação de dados ou informação de controle, sendo que o processamento lógico deve conter cálculo, ou criar dados derivados, ou manter ALI ou alterar o comportamento do sistema.



O processo para quantificação do sistema depende da documentação da fase de análise de requisitos, do tipo de contagem, do escopo da contagem, das funções de dados e de transação, do cálculo para determinar o tamanho funcional e a elaboração da documentação final da estimativa representados no fluxo da Figura 2.

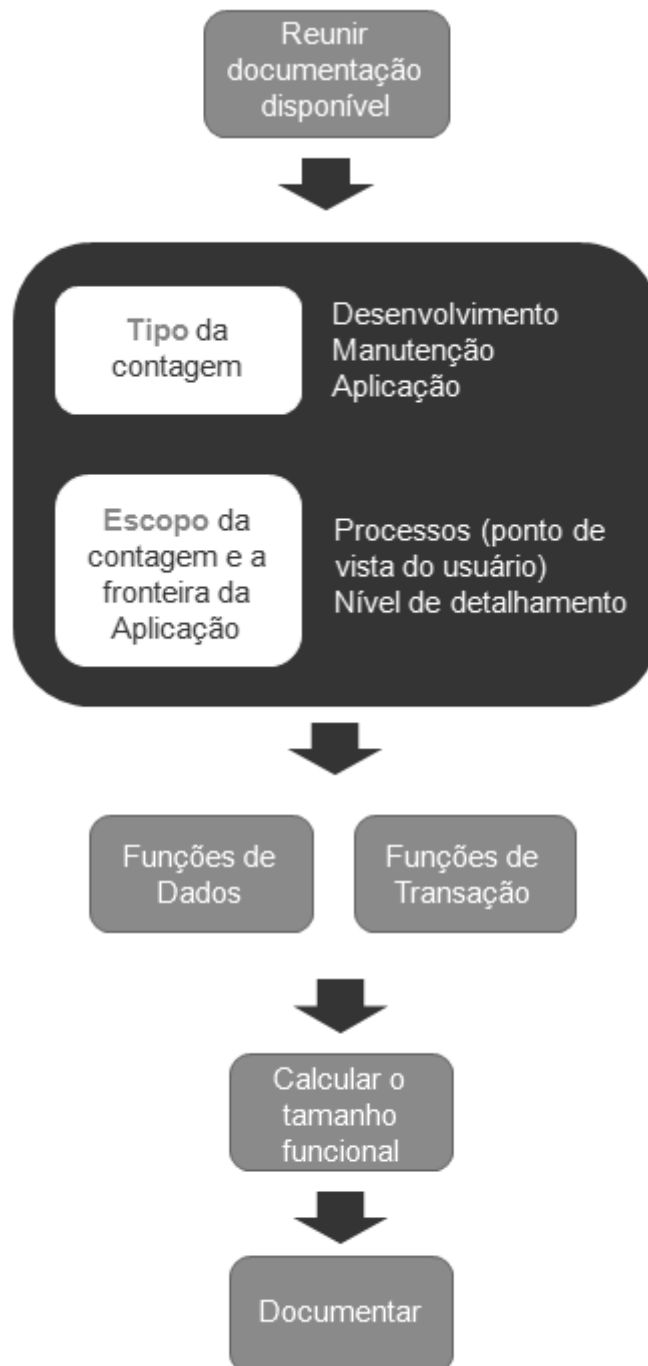


Figura 2 - Fluxo para contagem de Pontos de Funções ( Adaptado de IFPUG, 2010 )

Deve-se reunir a documentação disponível da fase de análise de requisitos para auxiliar na contagem. Geralmente os documentos disponíveis é o modelo de dados / objetos, diagrama de classes, diagrama do fluxo de dados, manuais do usuário, protótipos de telas e layouts de relatórios.

O tipo de contagem define se o projeto é um software novo considerado um projeto de desenvolvimento ou é um projeto de melhoria incluindo, excluindo e alterando novas funcionalidades ou é um projeto de aplicação relacionado a aplicação instalada consistindo de um ou mais componentes, módulos ou subsistemas.

O escopo da contagem e a fronteira de aplicação determinam os processos elementares baseados no ponto de vista do usuário e determinam o nível de detalhamento da contagem.

A complexidade funcional de cada função de dados deve ser determinada utilizando-se o número de Dados Elementares Referenciados (DERs) e Registros Lógicos Referenciados (RLRs), em conformidade com a Tabela 1.

Tabela 1 - Complexidade de Arquivos Lógicos Internos e Arquivos de Interface Externa (IFPUG, 2010)

		DERs		
		1 - 19	20 - 50	> 50
RLRs	1	Baixa	Baixa	Média
	2 – 5	Baixa	Média	Alta
	> 5	Média	Alta	Alta

A complexidade funcional de cada função de transação é determinada utilizando-se o número de ALRs e DERs, em conformidade com as Tabelas Tabela 2 e Tabela 3.

Tabela 2 - Complexidade funcional de Entradas Externas (IFPUG, 2010)

		DERs		
		1 - 4	5 – 15	> 15
ALRs	0 – 1	Baixa	Baixa	Média
	2	Baixa	Média	Alta

	<b>&gt; 2</b>	Média	Alta	Alta
--	---------------	-------	------	------

Tabela 3 - Complexidade de Consultas Externas e Saídas Externas (IFPUG, 2010)

		<b>DERs</b>		
		<b>1 - 5</b>	<b>6 – 19</b>	<b>&gt; 19</b>
<b>ALRs</b>	<b>0 – 1</b>	Baixa	Baixa	Média
	<b>2 – 3</b>	Baixa	Média	Alta
	<b>&gt; 3</b>	Média	Alta	Alta
NOTA Uma CE tem no mínimo 1 ALR.				

Cada função de dado e transação possui um peso em PF que é determinado pela complexidade das funções, respectivamente, representadas nas Tabelas Tabela 4 e Tabela 5.

Tabela 4 - Peso em PF baseado na Função de Dados x Complexidade (IFPUG, 2010)

		<b>Tipo</b>	
		<b>ALI</b>	<b>AIE</b>
<b>Complexidade Funcional</b>	<b>Baixa</b>	7	5
	<b>Média</b>	10	7
	<b>Alta</b>	15	10

Tabela 5 - Peso em PF baseado na Função de Transação x Complexidade (IFPUG, 2010)

		<b>Tipo</b>		
		<b>EE</b>	<b>SE</b>	<b>CE</b>
<b>Complexidade Funcional</b>	<b>Baixa</b>	3	4	3
	<b>Média</b>	4	5	4
	<b>Alta</b>	6	7	6

O cálculo dos pontos de função não ajustados é o produto do número de funções identificadas para uma determinada complexidade por sua contribuição e a somatória dos resultados são os pontos de funções não ajustados (PFNA).

A Tabela 6 representa um exemplo para o cálculo do PFNA gerados pelos ALI de um sistema hipotético e o mesmo deve ocorrer para outras funções do sistema (AIE, EE, SE e CE).

Tabela 6 - Exemplo de cálculo de pontos de função não ajustados

<b>Função</b>	<b>Itens por complexidade</b>	<b>Contribuição</b>	<b>Total por complexidade</b>	<b>Total de PFNA</b>
ALI	1 Baixa	X 7	7	47
	1 Média	X 10	10	
	2 Altas	X 15	30	

Os PFNA representam as funcionalidades que o sistema proporciona ao usuário mas não considera as especificidades do sistema, ou seja, o mesmo sistema pode ser implantado em um ambiente com arquitetura cliente servidor e em outro ambiente para funcionar em stand alone. Os requisitos funcionais são os mesmos mas as características dos sistemas variam entre os clientes. Para ajustar o nível de complexidade deve ser levado em consideração 14 características do sistema: comunicação de dados, processamento distribuído, performance, configuração intensamente utilizada, volume de transações, entrada de dados on-line, eficiência do usuário final, atualização on-line, processamento complexo, reusabilidade, facilidade de instalação, facilidade de operação, múltiplos locais e facilidade de mudança.

Deve-se atribuir níveis de influência para cada característica variando numa escala de 0 a 5, de nenhuma influência para total influência. A somatória dos níveis de influência será representada por TNI.

O valor fator de ajuste (VAF) é dado por:

$$VFA = (TNI * 0,01) + 0,65 \quad (1)$$

O total de pontos de função ajustados (PFA) é representado pela fórmula:

$$PFA = PFNA * VFA \quad (2)$$

Onde:

PFA é a contagem de pontos de função do projeto de desenvolvimento.

PFNA é a contagem de pontos de função não-ajustados para as funções que serão disponibilizadas depois da instalação.

VFA é o fator de ajuste.

Um documento deve ser elaborado com os resultados obtidos durante o processo conforme a Tabela 7:

Tabela 7 - Modelo de documentação da contagem de Pontos de Função para um processo

Requisito	Função	Tipo	AR	TD	Complex.	Contribuição
1.1	Relatório de Empresas	SE	1	5	Baixa	4 PF

### Pontos por Caso de Uso (PCU)

Criada em 1993 por Gustav Karner, os Pontos por Casos de Uso (PCU) permitem medir as funcionalidades do sistema baseado no modelo de caso de uso. (KARNER, 1993).

A métrica PCU é baseada na definição da Análise de Pontos por Função, a qual estima o tamanho do software na funcionalidade vista pelo usuário.

Os casos de uso principais do sistema devem ser identificados para estimar o tamanho do software como um todo.

Segundo Medeiros (2004), o processo de contagem dessa métrica é definido em 7 processos.

O processo 1 Identifica e classifica os atores de acordo com seu nível de complexidade, representados na Tabela 8, e calcula o TPNA (Total de Pesos não Ajustados dos Atores) somando os produtos da quantidade de atores pelo seu peso.

Tabela 8- Classificação dos Atores (MEDEIROS, 2004)

Complexidade do ator	Descrição	Peso
Simple	Muito poucas entidades de Banco de Dados envolvidas e sem regras de negócio complexas	1

Médio	Poucas entidade de Banco de Dados envolvidas e com algumas regras de negócio complexas	2
Complexo	Regras de negócios complexas e muitas entidades de Bancos de Dados presentes	3

O processo 2 conta os casos de uso e atribui o grau de complexidade baseada no número de classes e transações.

A transação é compreendida como uma série de processos que devem ser executados em conjunto. Caso não seja possível concluir o processamento, devem ser cancelados integralmente.

Deve ser calculado o TPNAUC (Total de Pesos não ajustados dos casos de usos) somando-se os produtos da quantidade de casos de usos pelo respectivo peso conforme Tabela 9.

Tabela 9 - Classificação dos Casos de Uso (MEDEIROS, 2004)

<b>Tipo de Caso de Uso</b>	<b>Descrição</b>	<b>Peso</b>
Simple	Considerar até 3 transações com menos de 5 classes de análise	5
Médio	Considerar de 4 a 7 transações com 5 a 10 classes de análise	10
Complexo	Considerar de 7 transações com pelo menos de 10 classes de análise	15

O calculo dos PCU's não ajustados é realizado no processo 3. Também são conhecidos como PCUNA e calculado através do somatório entre os pesos de atores (TPNAA) e casos de uso.(TPNAUC).

$$PCUNA = TPNAA + TPNAUC \quad (3)$$

O processo 4 determina o fator de complexidade técnica (FCT).

$$FCT = 0.6 + (0.01 * \text{Somatório do Fator técnico}) \quad (4)$$

Os fatores de complexidade técnica variam numa escala de 0 a 5, de acordo com o grau de dificuldade do sistema a ser desenvolvido. Deve ser considerado o valor 0 indicando que o requisito não está presente ou não é influente, o valor 3 indicando que o requisito possui influência média e o valor 5 indicando que o requisito possui influência significativa através de todo o processo.

O Somatório do Fator Técnico é composto pelo somatório do valor atribuído multiplicado pelo respectivo peso da Tabela 10.

Tabela 10 - Fatores de Complexidade Técnica (MEDEIROS, 2004)

<b>Descrição</b>	<b>Peso</b>
Sistemas Distribuídos	2,0
Desempenho da aplicação	1,0
Eficiência do usuário final (on-line)	1,0
Processamento interno complexo	1,0
Reusabilidade de código em outras aplicações	1,0
Facilidade de instalação	0,5
Usabilidade (facilidade operacional)	0,5
Portabilidade	2,0
Facilidade de manutenção	1,0
Concorrência	1,0
Características especiais de segurança	1,0
Acesso direto para terceiros	1,0
Facilidades especiais de treinamento	1,0

O Processo 5 determina o fator de complexidade ambiental (FCA).

$$FCA = 1,4 + (-0,03 * \text{Somatório do Fator Ambiental}) \quad (5)$$

Os fatores de complexidade ambientais indicam a eficiência do projeto e estão relacionados ao nível de experiência dos profissionais. Deve ser considerado o valor 0 indicando baixa experiência, o valor 3 indicando experiência média e o valor 5 indicando alta experiência.

O somatório do Fator Ambiental é composto pelo somatório do valor atribuído multiplicado pelo respectivo peso da Tabela 11.

Tabela 11 - Fatores de Complexidade Ambiental (MEDEIROS, 2004)

Fator	Descrição	Peso
F1	Familiaridade com o processo de desenvolvimento de software	1,5
F2	Experiência na aplicação	0,5
F3	Experiência com OO, na linguagem e na técnica de desenvolvimento	1,0
F4	Capacidade do líder de análise	0,5
F5	Motivação	1,0
F6	Requisitos estáveis	2,0
F7	Trabalhadores com dedicação parcial	-1,0
F8	Dificuldade da linguagem de programação	-1,0

O processo 6 determina os PCU's ajustados (PCUA) baseados nos PCU's não ajustados (PCUNA), complexidade técnica (FCT) e complexidade ambiental (FCA).

$$PCUA = PCUNA * FCT * FCA \quad (6)$$

O processo 7 calcula a estimativa de horas de programação.

Segundo KARNER, um PCU precisa de aproximadamente 20 horas/homem para ser concluídas, mas outros estudiosos sugerem o processo abaixo para obter a estimativa:

- X = total de itens de F1 a F6 com pontuação abaixo de 3
- Y = total de itens de F7 a F8 com pontuação acima de 3
- Se  $X + Y \leq 2$ , usar 20 como unidade de homens/hora
- Se  $3 \leq X + Y \leq 4$ , usar 28 como unidade de homens/hora
- Se  $X + Y \geq 5$ , deve-se tentar modificar o projeto de forma a baixar o número, pois o risco de insucesso é relativamente alto.

Estimativa de horas = PCUA \* pessoas/hora por unidade de PCU



## **Comparativo entre APF e PCU**

As métricas de APF e PCU proporcionam a estimativa do tamanho de um software baseado em fatores exigidos para cada uma.

A APF não depende exclusivamente dos Casos de Uso para aplicar a medição, pode utilizar o modelo de dados / objetos, diagrama de classes, diagrama do fluxo de dados, manuais do usuário, protótipos de telas e layouts de relatórios para auxiliar na contagem.

Pelo fato do PCU depender exclusivamente dos casos de uso é necessário que eles estejam atualizados, caso contrário, seria necessário escrevê-los para aplicar a medição. O fato de escrever os casos de uso apenas para a contagem torna a medição inviável considerando uma análise de custo x benefício. Com a APF é possível executar a medição analisando a própria aplicação em uso.

Ambas as métricas possuem um padrão definido para serem aplicadas, mas a PCU pode ter divergências nos resultados pois não existe um padrão único para elaborar os casos de uso.

As duas métricas não podem ser iniciadas antes da fase de análise dos requisitos ser finalizada.

A APF contempla projetos de desenvolvimento, melhoria e aplicações e o PCU contempla apenas projetos de desenvolvimento.

O PCU não possui grupos de usuários ou organização responsáveis pela padronização ou evolução da métrica como a APF que possui o IFPUG e é responsável por manter o Manual de Práticas de Contagem. (IFPUG, 2012).

Além do IFPUG, existem diversos fóruns de discussão sobre APF para troca de experiências e o BFPUG que é um grupo constituído com o objetivo de estimular e divulgar a utilização de métricas no desenvolvimento de sistemas. (BFPUG, 2012).

A APF é padronizada sob a norma ISO/IEC 20926 como um método de medição funcional aderente à ISO/IEC 14.143. (ISO, 2009).

O IFPUG possui um programa de certificação CFPS para a APF. (IFPUG, 2012).

A Tabela 12 representa os requisitos aplicáveis nas métricas de APF e PCU.

Tabela 12 - Requisitos aplicáveis em APF e PCU

Item	Requisitos	APF	PCU
1.1	Depende exclusivamente da documentação de Casos de Uso.	NÃO	SIM
1.2	Necessita que a documentação esteja atualizada.	NÃO	SIM
1.3	Possui um padrão único para que a métrica seja aplicada evitando divergências nos resultados.	SIM	NÃO
1.4	Pode ser iniciado antes da fase de análise dos requisitos ser finalizada.	NÃO	NÃO
1.5	Contempla projetos de Desenvolvimento.	SIM	SIM
1.6	Contempla projetos de Melhoria (manutenção).	SIM	NÃO
1.7	Contempla projetos de Aplicações.	SIM	NÃO
1.8	Possui grupo de usuários ou organização responsável pela padronização ou evolução.	SIM	NÃO
1.9	Adere a norma ISO/IEC 14143 - Define um modelo para a medição funcional de software.	SIM	NÃO
1.10	Possui um programa de certificação de profissionais.	SIM	NÃO
1.11	Possui requisitos que influenciam na contagem entre organizações possibilitando que a contagem se torne variável.	NÃO	SIM

### Aplicação das métricas APF e PCU

Para demonstração da aplicabilidade das duas métricas iremos utilizar um exemplo simples.

A empresa X precisa implementar um módulo para cadastro de usuários em seu sistema e solicitou uma estimativa do tamanho a duas fábricas de software, respectivamente, empresa Y e empresa Z.

A única documentação disponibilizada as empresas, foi o caso de uso apresentado no Apêndice A.

A empresa Y utiliza a APF, distribuindo-a em cinco processos, como padrão para estimar suas demandas.

O processo 1 é a identificação do escopo da contagem e fronteira de aplicação. A solicitação da empresa X é um projeto de desenvolvimento para fins de contagem, pois novas funcionalidades estão sendo incorporadas. Por questões de

simplicidade, não farão parte do escopo da contagem todas as outras funções do sistema. A fronteira da aplicação quanto a medição do caso real serão limitados, para fins didáticos, ao cadastro de Usuários, objeto do estudo de caso.

O processo 2 representa a contagem das funções de dado. Deve-se identificar o modelo de dados, representado na Figura 3, para contagem dos TDs e ARs e apresentados na Tabela 13 como 5 TDs (Nome, CPF, Área, Telefone, Status) e 1 AR (Usuário).



Figura 3 - Grupo de dados relativos ao cadastro do Usuário

Tabela 13 - Funções de Dados

Função de Dados	Tipo	ARs	TDs	Complexidade
Usuario	ALI	1	5	Baixa

O processo 3 é responsável pela contagem das funções de transação. A partir das telas apresentadas no Apêndice A e do modelo de dados representados na Figura 3 foram identificadas as funções de transação apresentadas na Tabela 14.

Tabela 14 - Funções de Transação

Função de Transação	Tipo	ARs	TDs	Complexidade
Incluir Usuário	EE	1	5	Baixa
Editar Usuário	EE	1	6	Baixa
Pesquisar Usuário	CE	1	6	Baixa

O processo 4 calcula o valor de ajuste para os pontos de função. Deve-se preencher a Tabela 15 identificando a somatória dos níveis de influência, representados por TNI, e aplicar o resultado na fórmula (1).

Tabela 15 - Tabela com fatores de ajuste

Fatores de Ajuste		
	Características do Sistema	Influencia
1	Comunicação de Dados	3
2	Funções Distribuídas	3
3	Desempenho	3
4	Configuração de Equipamentos	3
5	Volume de Transações	3
6	Entrada de Dados Online	3
7	Interface com usuário	3
8	Atualização On-line	2
9	Processamento Complexo	2
10	Reusabilidade	2
11	Facilidade de implantação	2
12	Facilidade Operacional	2
13	Múltiplos Locais	2
14	Facilidade de mudanças (Flexibilidade)	2
<b>Total (TNI)</b>		<b>35</b>
<b>VFA = (TNI * 0,01 + 0,65)</b>		<b>1</b>

O processo 5 apresenta o cálculo do valor ajustado do ponto de função. Deve-se aplicar a fórmula (2) aos seus respectivos valores de ponto de função obtidos através da complexidade da Tabela 13 e Tabela 14 com a Tabela 2, Tabela 3 e Tabela 4, representados na Tabela 16.

Tabela 16 - Cálculo do valor ajustado do ponto de função

Função de Transação / Dados	Tipo	ARs	TDs	Complexidade	PF
Incluir Usuário	EE	1	5	Baixa	3
Editar Usuário	EE	1	6	Baixa	3
Pesquisar Usuário	CE	1	6	Baixa	3
Usuario	ALI	1	5	Baixa	7
PFNA					16
PFA = PFNA * VAF = 16 * 1					16

Considerando-se uma produtividade de 8hrs/PF, a empresa Y apresentou a estimativa de 128 horas para desenvolvimento das novas funcionalidades à empresa X.

A empresa Z utiliza o PCU, distribuindo-o em sete processos, como padrão para estimar suas demandas.

O processo 1 Identifica e classifica os atores de acordo com seu nível de complexidade. Considerando o caso de uso do Apêndice A e a Tabela 8 foi identificado um ator de complexidade simples e peso 1. O TPNA (Total de Pesos não Ajustados dos Atores) é a somatória dos produtos da quantidade de atores pelo seu peso, logo o TPNA possui o valor de 2.

O processo 2 conta os casos de uso e atribui o grau de complexidade baseada no número de classes e transações. O cálculo de TPNAUC (Total de Pesos não ajustados dos casos de usos) é a somatória dos produtos da quantidade de casos de usos pelo respectivo peso conforme Tabela 9. O caso de uso do Apêndice A foi classificado como Médio e peso 5, logo o TPNAUC obteve o valor de 6.

O processo 3 calcula os PCU's não ajustados ou PCUNA através do somatório entre os pesos de atores (TPNA) e casos de uso (TPNAUC) através da fórmula (3), logo o seu valor é de 8.

O processo 4 determina o fator de complexidade técnica (FCT) calculado pela fórmula (4) e representado na Tabela 17.

Tabela 17 - Tabela de cálculo do Fator de Complexidade Técnica

<b>Complexidade Técnica</b>				
<b>Fator</b>	<b>Descrição</b>	<b>Peso</b>	<b>Valor</b>	<b>Fator</b>
CT1	Sistema Distribuído	2	3	6
CT2	Necessidade de performance	1	3	3
CT3	Interface com usuário final	1	3	3
CT4	Complexidade do processo	1	3	3
CT5	Código reutilizável	1	3	3
CT6	Facilidade de instalação	0,5	3	1,5
CT7	Facilidade do uso	0,5	3	1,5
CT8	Portabilidade	2	3	6
CT9	Facilidade para alteração	1	3	3
CT10	Concorrência	1	3	3
CT11	Segurança	1	3	3
CT12	Acesso a outros sistemas	1	3	3
CT13	Necessidade de treinamento	1	3	3
<b>Total Complexidade Técnica</b>				<b>42</b>
<b>FATOR COMPLEXIDADE TÉCNICA</b>				<b>1,02</b>

O Processo 5 determina o fator de complexidade ambiental (FCA) calculado pela fórmula (5) e representado na Tabela 18.

Tabela 18 - Tabela de cálculo do Fator de Complexidade Ambiental

<b>Complexidade de Ambiente</b>				
<b>Fator</b>	<b>Descrição</b>	<b>Peso</b>	<b>Valor</b>	<b>Fator</b>
CA1	Familiaridade com o processo de desenvolvimento	1,5	3	4,5
CA2	Experiência com a aplicação	0,5	3	1,5
CA3	Experiência com orientação a objetos	1	3	3
CA4	Presença de Analista experiente na equipe do projeto	0,5	3	1,5
CA5	Motivação	1	3	3
CA6	Estabilidade dos requisitos	2	3	6
CA7	Equipe part-time / varios times	-1	3	-3
CA8	Dificuldade da linguagem de programação	-1	3	-3
<b>Total Complexidade Ambiente</b>				<b>13,5</b>
<b>FATOR COMPLEXIDADE AMBIENTE</b>				<b>0,995</b>

O processo 6 determina os PCU's ajustados (PCUA) baseados nos PCU's não ajustados (PCUNA), complexidade técnica (FCT) e complexidade ambiental (FCA) através do seu produto representado na fórmula (6), obtendo o valor de 8,1192.

O processo 7 calcula a estimativa de horas de programação, sendo que a empresa Z utiliza uma produtividade de 20hrs/PCUA e apresentou sua estimativa de 162,384 horas para desenvolvimento das novas funcionalidades à empresa X.

## **Considerações Finais**

A APF é uma métrica que exige maior domínio de seus requisitos para proporcionar uma melhor quantificação de um sistema. A diversidade de documentos que podem ser utilizados para auxiliar na técnica proporciona maior detalhamento para descrever e calcular os pontos de funções. A visão dos processos documentados é de fácil entendimento pelo usuário pois todos os processos são visíveis ao mesmo.

Os Casos de Uso não possuem um modelo padrão para serem escritos e o PCU pode variar consideravelmente ao ser elaborados em organizações distintas pois é baseado exclusivamente nos Casos de Uso do sistema.

O número de especialistas em Análise de Pontos de Função é consideravelmente baixo no Brasil pois o índice de demandas para desenvolvimento e manutenção de sistemas por organizações públicas e privadas vem aumentando nos últimos anos fazendo com que estes serviços sejam terceirizados. O exame para certificação é complexo, o que impede de existirem muitos especialistas certificados nesta técnica.

O aumento nas demandas exige que um controle mais rigoroso seja aplicado nos orçamentos públicos que seguem a normativa IN04/2010 obrigando o uso de métricas nos contratos de serviços relacionados a softwares.

Pode ser elaborado uma conversão entre a APF e o PCU baseado nos requisitos solicitados pelo cliente e, no caso de licitações, nos editais propostos pelas organizações.

Considerando o comparativo de APF e PCU podemos concluir que a APF é consideravelmente superior ao PCU por não precisar exclusivamente dos casos de uso e proporciona ao cliente uma maior segurança relacionado a quantificação do sistema pois possui um padrão definido e mantido por uma organização internacional.

## REFERÊNCIAS

APF METRICAS. Usuários, 2012. Disponível em: <<http://www.apfmetricas.com.br/usuario-metricas.html>>. Acesso em: 14 dez. 2012.

BERTOLAMI, M. A. Una Propuesta de Análisis de Puntos Función aplicado a LEL y Escenarios. Facultad de Informática, Universidad Nacional de La Plata, Argentina, 2003.

BFPUG. O que é BFPUG. , 2012. Disponível em: <<http://www.bfpug.com.br>>. Acesso em: 10 nov. 2012.

FENTON, N., PFLEEGER, S. Software Metrics, A rigorous and Practical Approach. Second Edition, PWS Publishing Company, 1996.

IFPUG. About function point analysis, 2012. Disponível em: <[http://www.ifpug.org/?page\\_id=10](http://www.ifpug.org/?page_id=10)>. Acesso em: 10 nov. 2012.

IFPUG. CFPS Certification Extension Program, 2012. Disponível em: <[http://www.ifpug.org/?page\\_id=312](http://www.ifpug.org/?page_id=312)>. Acesso em: 10 nov. 2012.

IFPUG, Manual de Práticas de Contagem de Pontos de Função, versão 4.3.1 (Traduzida). Westerville, Ohio (International Function Point Users Group (IFPUG), 2010. ISBN 978-0-9753783-5-9.

ISO. ISO/IEC 20926:2009 - Software and systems engineering -- Software measurement -- IFPUG functional size measurement method 2009. Disponível em: <[http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=51717](http://www.iso.org/iso/catalogue_detail.htm?csnumber=51717)>. Acesso em: 09 dez. 2012.

JONES, Capers. Applied Software Measurement. 3 ed. United States: MC Graw Hill, 2008.

KARNER, Gustav. Resource estimation for objectory projects. Objective systems SF AB. Kista, 17 set. 1993.

MEDEIROS, Ernani. Desenvolvendo software com uml 2.0. São Paulo: Makron Books, 2004.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software. 12. ed. São Paulo: Érica, 2012.



## Apêndice A – Caso de Uso Manter Usuário

### Descrição Resumida

Este caso de uso descreve as funcionalidades de pesquisa, inclusão e alteração de cadastro de usuários.

### Ator Principal

1. Usuário.

### Ativação

Este caso de uso deve iniciar quando:

1. Usuário acessa no menu “Usuário”.

### Pré-Condições

Estar cadastrado no sistema e possuir acesso para o cadastro de usuários.

### Pós-condições

Usuário cadastrado com sucesso.

### Fluxo Principal

1. O usuário acessa o menu Usuário.
2. O sistema carrega a tela “Pesquisar Usuário”. **(T1)(A1)(A2)**
3. O usuário clica no botão “Novo”.
4. O sistema carrega a tela para inclusão de usuários. **(T2)**
5. O usuário preenche as informações clica em “Salvar”. **(E1)**
6. O caso de uso é encerrado.

### Fluxo Alternativo

**A1:** Pesquisar usuários cadastrados.

1. O usuário preenche os campos e clica em “Pesquisar”. **(R1)(E2)**
2. O sistema apresenta as informações na mesma tela.

**A2:** Alterar o cadastro de um usuário já cadastrado.

1. O usuário pesquisa o cadastro do usuário e clica no ícone de edição. **(A1) (T2)**.
2. O sistema apresenta a tela para alteração das informações do usuário. **(T3)**
3. O usuário altera as informações e clica em “Salvar”. **(E1)**
4. O sistema retorna a mensagem “Dados alterados com sucesso!”.

### Fluxo Exceção

**E1:** Campos Obrigatórios

1. Os campos “Nome”, “CPF”, “Área” e “Telefone” são de preenchimento obrigatório.

**E2:** Usuário não encontrado

1. O sistema não localiza o usuário pelo filtro selecionado e exibe a mensagem: “Não existem informações disponíveis”.

## REGRAS

**R1:** O sistema listará todos os usuários cadastrados no sistema.

## TELAS

**T1:** Tela para pesquisar usuários

The screenshot shows a web browser window titled "A Web Page". The address bar contains "http://". The page has a navigation bar with three buttons: "Usuário" (highlighted in blue), "Relatório", and "Sair". Below the navigation bar, there are input fields for "Nome" and "CPF", and dropdown menus for "Área" (with "Selecione" as the selected option) and "Status" (also with "Selecione" as the selected option). There is also a "Telefone" input field. Below these fields are two buttons: "Pesquisar" and "Novo". At the bottom, there is a table with the following data:

Nome	CPF	Área	Telefone	Status	Editar
João	111.111.111-11	Informática	(11) 1111-1111	Ativo	<input type="checkbox"/>
José	222.222.222-22	RH	(22) 2222-2222	Inativo	<input type="checkbox"/>
Maria	333.333.333-33	RH	(33) 3333-3333	Ativo	<input type="checkbox"/>

**T2:** Tela para inclusão de usuário

The wireframe depicts a web browser window with a title bar 'A Web Page'. The address bar contains 'http://'. The browser's navigation buttons (back, forward, stop, home) are visible. A menu bar at the top contains three items: 'Usuário' (highlighted in blue), 'Relatório', and 'Sair'. Below the menu bar is a form titled 'Incluir Usuário'. The form contains the following fields: a text input for 'Nome', text inputs for 'CPF' and 'Telefone', and a dropdown menu for 'Área' with the text 'Selecione'. A 'Salvar' button is positioned at the bottom left of the form. The browser's status bar at the bottom right shows a double-slash icon.

**T1:** Tela para edição de usuário

The wireframe depicts a web browser window with a title bar 'A Web Page'. The address bar contains 'http://'. The browser's navigation buttons (back, forward, stop, home) are visible. A menu bar at the top contains three items: 'Usuário' (highlighted in blue), 'Relatório', and 'Sair'. Below the menu bar is a form titled 'Editar Usuário'. The form contains the following fields: a text input for 'Nome', text inputs for 'CPF' and 'Telefone', a dropdown menu for 'Área' with the text 'Selecione', and another dropdown menu for 'Status' with the text 'Selecione'. A 'Salvar' button is positioned at the bottom left of the form. The browser's status bar at the bottom right shows a double-slash icon.